



# Dreambuilder Bot

*Where we build Dreams and MVPs*

By: Adam, Faith, and Yen

# Agenda

- Problem Statement
- Approach
- Model Selection
- Data Cleanup & Model Training
- Model Evaluation
- Findings
- Conclusion
- Postmortem: Challenges
- Postmortem: Next Steps



# Problem Statement

- **Problem Statement:** The Crypto market is known to be extremely volatile and hard to predict. It can play with your emotions causing you to FOMO into the top or sell into extreme fear at the bottom. Can machine learning help us decide when to make our trades?
- **Objective:** to identify a type of analysis and/or indicators that predict the price movement of a crypto and apply different machine learning models to maximize returns.
- **Key Questions:**
  - Which analysis is better to predict price movement for cryptos: technical or sentiment?
  - Can we combine sentiment and technical analysis with machine learning to build a profitable bot?
  - Which machine learning model will optimize the results?



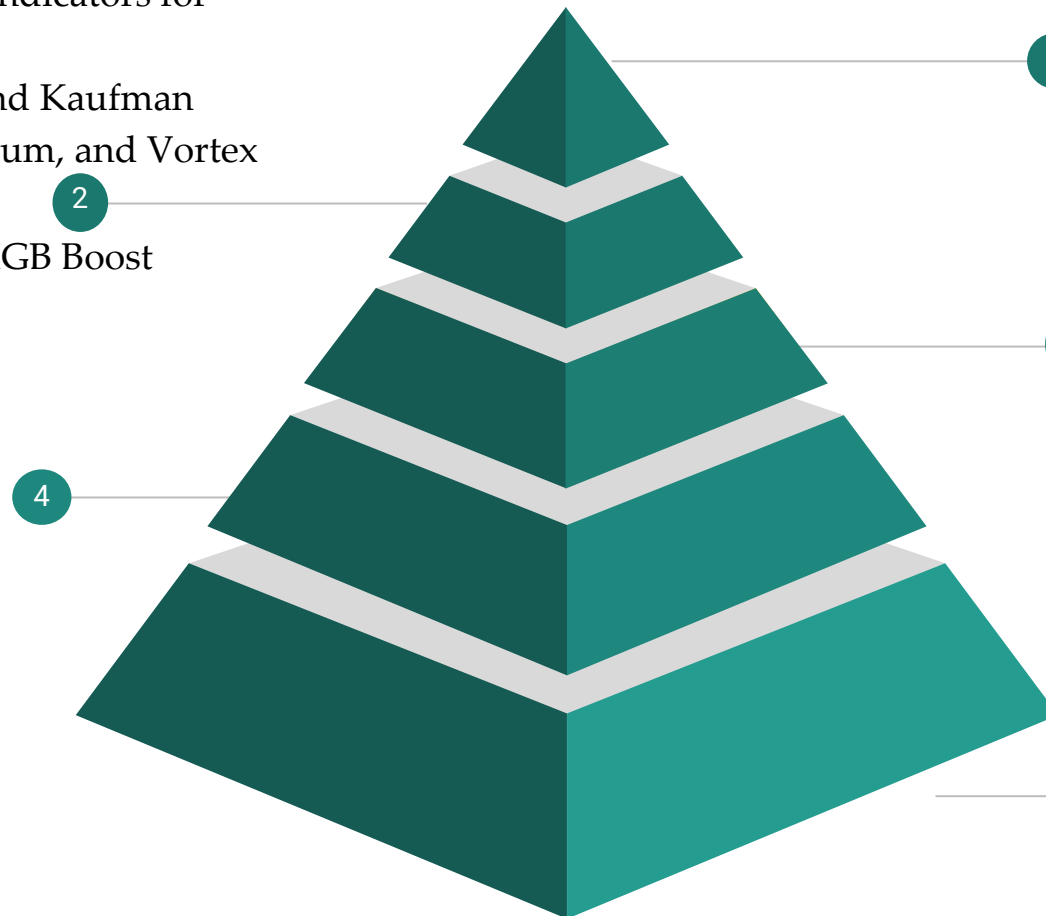
# Approach

## Selection & Data Collection

Select BTC and ETH for cryptos  
Select tweets, pytrend, fear & greed indicators for Sentiment analysis  
Select candlestick pattern, Chaikin and Kaufman Efficiency Indicator, Market Momentum, and Vortex indicators  
Select SVM, Neural Networks, and XGB Boost

## Analysis

Analyze results  
Compile report



## Research

Research API for Sentiment analysis  
Research Technical analysis libraries  
Research live stream data  
Research Machine Learning Models

## Exploration/Data Clean up/Training

Build new environment with Python 3.8  
Data Clean up  
Apply the sentiment and technical indicators  
Perform Backtesting  
Build two bots for trading crypto. Bot A uses sentiment analysis from the Fear and Greed index and trades Ethereum. Bot B uses multiple technical indicators and trades ETH

## Present Project

Here we are

# Model Selection

Predictive Model	Pro	Con
<b>Support Vector Machines (SVM)</b> is a supervised learning model that we can use for classification and regression analysis. SVM separates classes of data points into multidimensional space.	Performs very well with a limited amount of data to analyze.	Model worked well in market environment similar to training data. Failed in volatile market with short training period.
<b>Multilayer Perceptron Classifier (MLP Classifier)</b> is a technique of feed-forward artificial neural networks using a back propagation learning method to classify the target variable used for supervised learning.	It uses to solve complex nonlinear problems, and works well with large input data with a relatively faster performance. The algorithm tends to achieve the same accuracy ratio even with smaller data.	When there is a complex data set it is hard to figure out the issues.
<b>Logistic Regression</b> is a supervised machine learning model. It finds the best fit logistic function to describe the relationship between input variables (X) and a categorical output variable (y).	Simple to implement Feature scaling not needed: Does not require input features to be scaled	Poor performance with irrelevant and highly correlated features (use Boruta plot for removing similar or correlated features and irrelevant features). High reliance on proper presentation of data. All the important variables / features should be identified for it to work well.
<b>XGBoost</b> is an ensemble learning algorithm that make predictions using Decision Trees.	It uses Classification and Regression (CART) trees instead of containing a decision in each leaf. Less feature engineering required (No need for scaling, normalizing data, can also handle missing values well)	It doesn't work well on sparse data. Overfitting possible if parameters not tuned properly

# Data Cleanup & Model Training

## Data Collection

- **Yahoo Finance:** Historical crypto price data (Daily timeframe)
- **Requests library:** to access API for Fear and Greed value
- **FinTA library:** easy to implement technical indicators
- **Pytrends:** daily reading of google searches for different cryptocurrencies
- **Tweepy:** Library to access twitter API for sentiment analysis.
- **Japanese Candlestick library:** for advanced technical indicators (Hammer & Doji)
- **Alternative.me:** API used to pull Fear and Greed value
- **Websocket-client :** for pulling live crypto prices (every 10 seconds). Worked with the **Binance API** and **Alpaca API**

## Data Preprocessing

- **Pandas.DataFrame.to\_csv:**
- **Pandas.read\_csv:**  
for easy access to dataframes
- **Pandas.get\_dummies:** to  
convert categorical variable into  
dummy/indicator variable (1.0  
long or -1.0 short)
- **StandardScaler:** to scale  
features for machine learning
- **DateOffset:** to split training  
and testing data
- **Pipeline** to Transform Data
- **Convert** selected value of data

## Machine Learning

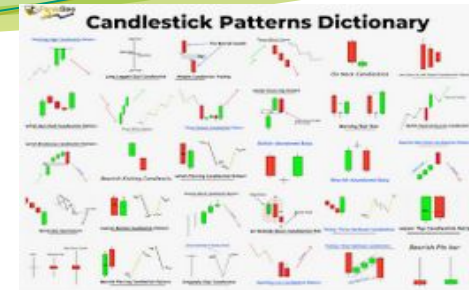
- **Support Vector Machines (SVM):**  
Supervised ML model that uses  
classification algorithms for two  
group classification. (better  
performance with limited number of  
samples)
- **Logistic Regression:** supervised ML  
model. It finds the best fit logistic  
function to describe the relationship  
between input variables (X) and a  
categorical output variable (y).
- **XGBoost:** Works by training a  
number of decision trees. each tree  
trained on a subset of data, the  
predictions are then combined to  
form the final prediction.

# Fear and Greed Strategy

- Algorithm uses the daily Fear and Greed reading to determine whether to go long or short on a cryptocurrency.
- “Each day, we analyze emotions and sentiments from different sources and crunch them into one simple number: The Fear and Greed Index”
- “Be fearful when others are greedy”
- “Be greedy when others are fearful”
- **Logic:**
- $FG \geq 95$ : -1.0 (Short)
- $95 > FG \geq 46$ : 1.0 (Long)
- $46 > FG \geq 22$ : -1.0 (Short)
- $FG < 22$ : 1.0 (Long)



# Technical Analysis Strategy



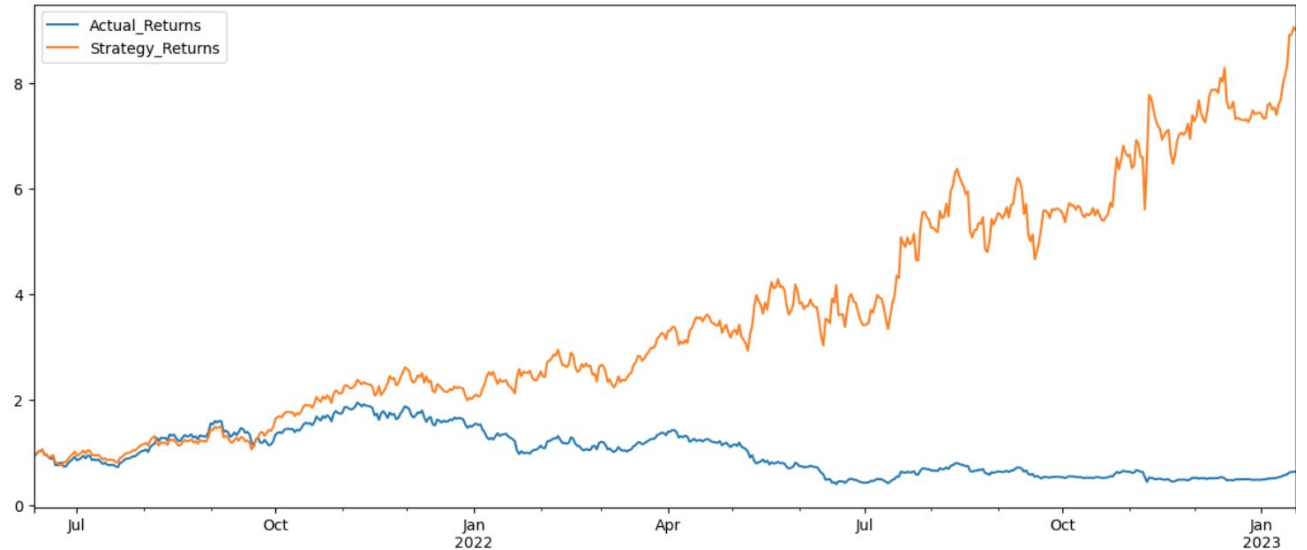
- **Candlestick patterns:** Use hammer candlesticks to enter a position. It indicates a potential price reversal to the upside. Use Doji candlesticks to exit a position. It indicates bearish reversal over the near term following the breakout
- **Market momentum:** the aggregate rate of acceleration for the broader market as a whole. It indicates if the trend is sustainable in the future. If MoM is greater than 0 then it is bullish and MoM is less than 0 it is bearish.
- **Chaikin Oscillator:** measure the accumulation-distribution line of moving average convergence-divergence (MACD). It indicates if institutions/whales are buying or selling.
- **Kaufman Efficiency Ratio (ER):** provide a method of quantifying a market's noise. It assists with identifying false break out.
- **Vortex:** spot trend reversals and confirm current trend by using a pair of oscillating lines. One to identify positive trend movement and the other to identify negative price movement. When the two crosses between the lines trigger buy and sell signals that are designed to capture the most dynamic trending action, higher or lower.
- **Relative Strength Index (RSI):** measure the speed and magnitude of the recent price changes to evaluate overvalued or undervalued conditions in the price. Overvalued > 70% and Undervalued < 30%



# Model Evaluation

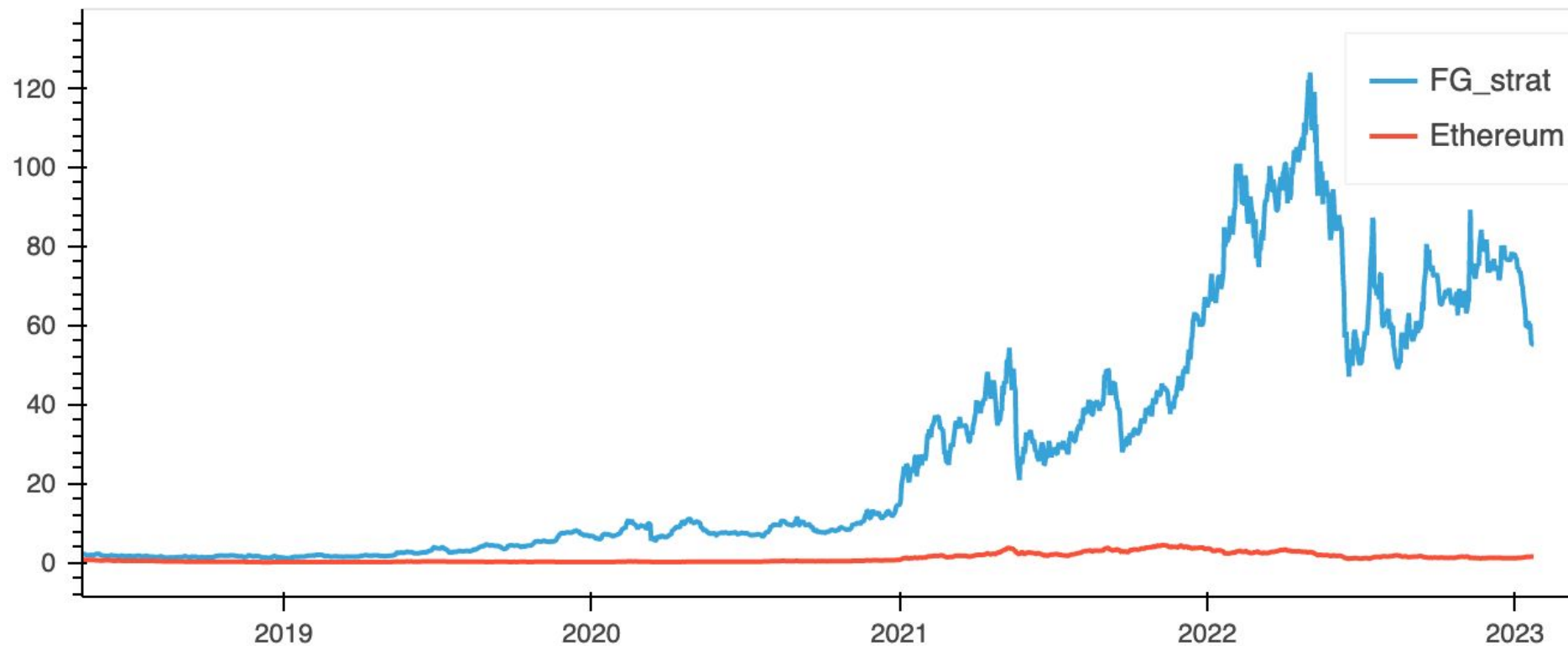
- Examine ROI and confusion matrix/classification report.

	precision	recall	f1-score	support
-1.0	0.54	0.52	0.53	589
1.0	0.56	0.58	0.57	628
accuracy			0.55	1217
macro avg	0.55	0.55	0.55	1217
weighted avg	0.55	0.55	0.55	1217



# F&G strategy results

- **Backtesting results:  
(5 years Ethereum)**
- The F&G strategy was able to outperform ETH by a large margin
- It is important to remember that past performance does not guarantee future results



start of historical  
data: 02-03-2018

Buying \$1000 and holding: \$1513.9

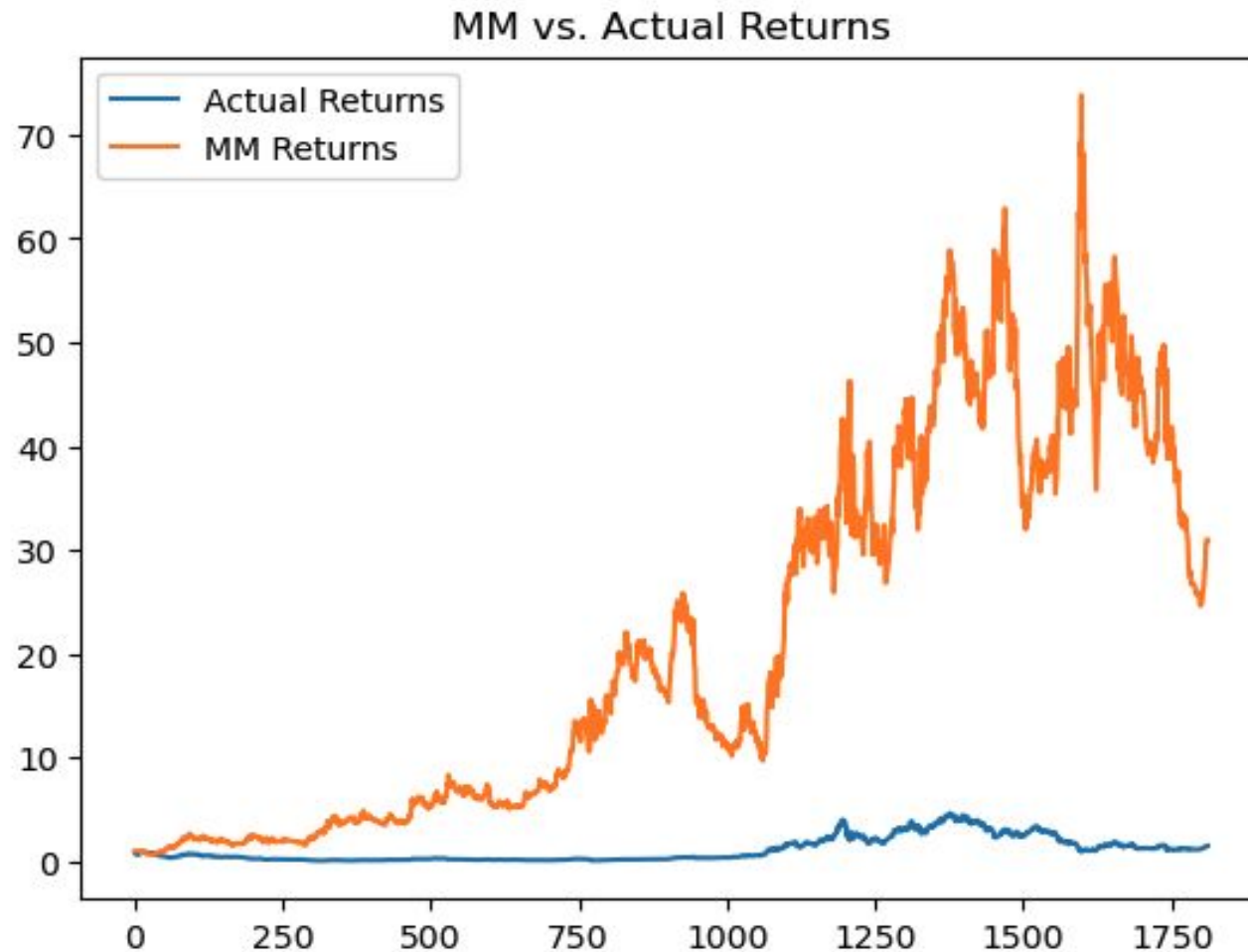
\$1000 with fear and greed strategy: \$55124.5

# Sharing Code

- Fear and Greed Strategy with perfect signal as target (Final\_Machine\_Learning) - Adam
- SVM with MoM indicator (Tech\_FG\_Model\_Training) - Faith
- XGB Model with trade execution (Yen\_Alpacamarkets) - Yen

# MoM results

- **Backtesting results:  
(5 years Ethereum)**
- Market Momentum returned 30x in 4 years
- We continued to refine the models as they were built. The Fear and Greed strategy outperformed MoM strategy, but the returns for both were outstanding: 30x in 4 years using Market Momentum, (55x) in 4 years using F&G index.



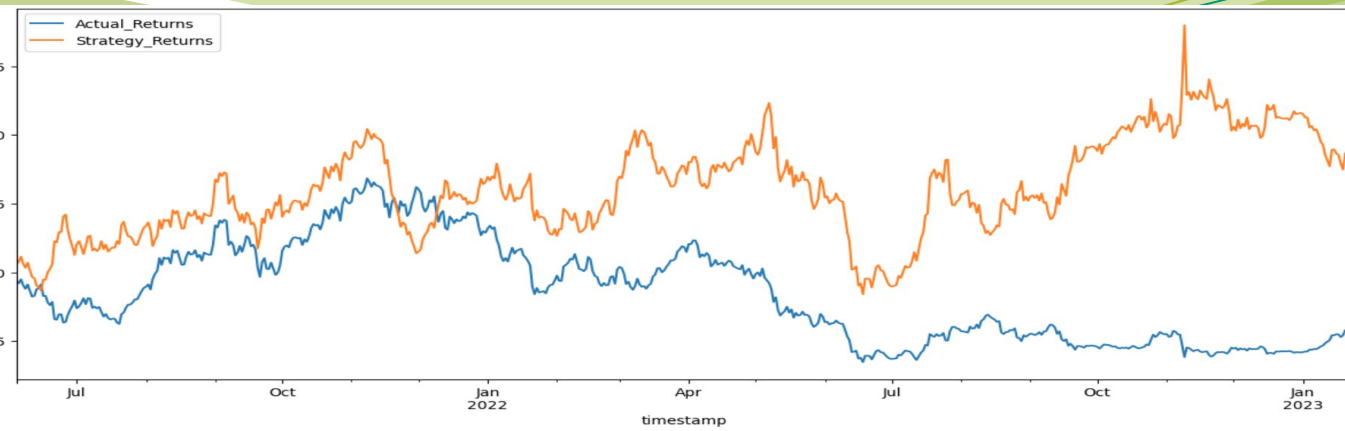
buying \$1000 and holding: \$1512.21

buying \$1000 with Market Momentum strategy: \$30872.24



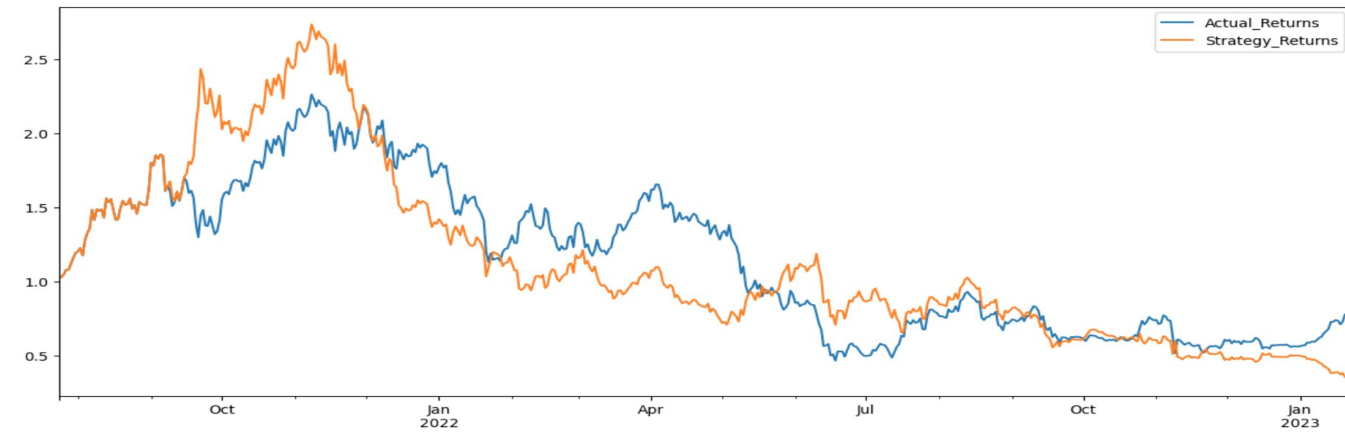
## F&G Features (Sentiment only):

	precision	recall	f1-score	support
-1.0	0.53	0.44	0.48	587
1.0	0.55	0.63	0.58	627
accuracy			0.54	1214
macro avg	0.54	0.53	0.53	1214
weighted avg	0.54	0.54	0.53	1214



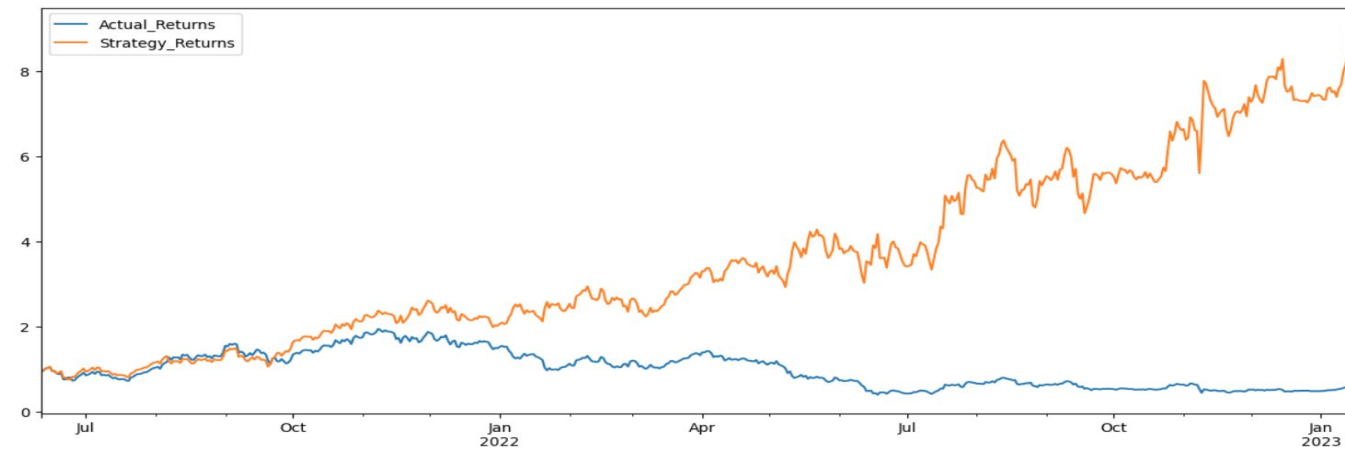
## F&G AND TA Features:

	precision	recall	f1-score	support
-1.0	0.53	0.45	0.49	585
1.0	0.55	0.64	0.59	631
accuracy			0.55	1216
macro avg	0.54	0.54	0.54	1216
weighted avg	0.54	0.55	0.54	1216



## MoM Features (TA only):

	precision	recall	f1-score	support
-1.0	0.54	0.52	0.53	589
1.0	0.56	0.58	0.57	628
accuracy			0.55	1217
macro avg	0.55	0.55	0.55	1217
weighted avg	0.55	0.55	0.55	1217



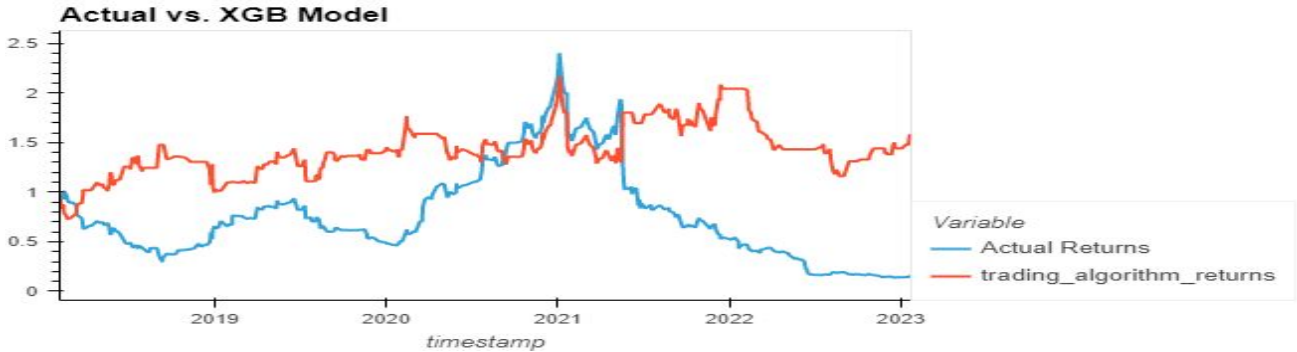
Logistic  
Regression  
with Fear  
and Greed  
& MoM

	precision	recall	f1-score	support
-1.0	0.79	0.62	0.69	738
1.0	0.73	0.86	0.79	890
accuracy			0.75	1628
macro avg	0.76	0.74	0.74	1628
weighted avg	0.76	0.75	0.75	1628



XGBoost  
with Fear  
and Greed  
& RSI

	precision	recall	f1-score	support
0	0.84	0.84	0.84	123
1	0.89	0.93	0.91	162
2	0.81	0.78	0.79	169
accuracy			0.85	454
macro avg	0.84	0.85	0.85	454
weighted avg	0.84	0.85	0.84	454



# Results(Machine Learning)

## Target = perfect signal

- SVM with **Fear and Greed** strategy as the only feature (sentiment only)  
**Returns** over testing period: **1.68**  
**Accuracy: 0.54**
- SVM with **Fear and Greed signal AND technical indicators** as features  
**Returns** over testing period: **0.35**  
**Accuracy: 0.55**
- SVM with **Market Momentum signal** (TA only)  
**Returns** over testing period: **9.0**  
**Accuracy: 0.55**

## Target = MM signal

- Logistic Regression for **MM strategy** with other technical strategies (Chaikin, ER, Vort -Di, Vort +Di) and Fear and Greed as features  
**Returns** over testing period: **8.46**  
**Accuracy: 0.75**

## Target = Fear & Greed

- XGBoost Model with Fear & Greed & RSI as features  
**Returns** over testing period: **1.58**  
**Accuracy: 0.85**

# Conclusion



- At every fork in the road, we kept moving forward with MVP, but left lots of successful strategies to be explored.
- Both technical and sentiment analysis can be used with machine learning to build trading bots with high ROIs
- From our results, we recommend using either technical or sentiment analysis but not both.



# Postmortem: Challenges

- Certain Python libraries work on Python 3.8 or better
- Tweepy - only 7 days of past data available at access level. Would need to record and test data daily to build model
- Pytrends - need more research for daily information. We already had two successful strategies and moved forward, but there is more to be researched here.
- Livestream Data issue - unable to save the object into a dataframe to feed into machine learning model.
- When to use the .shift() function
- SLTP Ratio - trying to find the actual trend on the monthly

# Postmortem: Next Steps

- More analysis on correlations
- Address some challenges that we have experienced (Tweepy, Pytrends, Livestream data, and SLTP ratio)
- Bot maintenance and notifications
- Run other Machine Learning Models
- Expand to stocks and incorporate fundamental analysis

